

# Human-Friendly Arm Robot Berbasis Interactive Particle Swarm Optimization (IPSO)

Anhar Risnumawan<sup>1</sup>, Indra Adji Sulistijono<sup>1</sup>

<sup>1</sup>Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya  
Kampus PENS-ITS Sukolilo, Surabaya  
[Antong@student.eepis-its.edu](mailto:Antong@student.eepis-its.edu)  
[indra@eepis-its.edu](mailto:indra@eepis-its.edu)

**Abstrak** Makalah ini membahas tentang lengan robot sebagai pembangkit trayektori untuk jabat tangan dengan manusia. Berbagai metode pembangkit trayektori telah dikenalkan, tetapi robot harus dapat berinteraksi dengan lingkungan termasuk dengan manusia. Dalam makalah ini diterapkan *interactive particle swarm optimization* sebagai pembangkit trayektori menggunakan evaluasi manusia. Dibutuhkan suatu metode untuk memperkirakan nilai evaluasi manusia, untuk itu digunakan fungsi *state-value*. Hasil percobaan menunjukkan trayektori terbaik diperoleh saat *trial* kelima dengan parameter pemberat jumlah sudut, jumlah jarak, dan jumlah evaluasi manusia yaitu 0.3, 0.1, dan 0.6.

**Kata kunci**— Robot, trayektori, *interactive particle swarm optimization*, evaluasi, *state-value*

## I. PENDAHULUAN

Dewasa ini robot *human-friendly* dan robot partner telah banyak dikembangkan untuk generasi selanjutnya demi kepentingan manusia [1]-[3]. Robot – robot tersebut memerlukan kemampuan kecerdasan melalui interaksi antara robot dengan manusia. Robot harus dapat menghasilkan tingkah laku yang diinginkan sebagai komunikasi dengan pemiliknya [1]-[3]. S.Kristensen dkk, mengenalkan metode pembelajaran secara langsung antara manusia dengan robot dalam aplikasi pelayan kopi [3]. Artur M.A. mengenalkan strategi – strategi untuk memperoleh kemampuan tertentu berbasis interaksi robot dengan manusia. Interaksinya dengan manusia memberikan kemampuan baru sebagai seorang aktor [4].

Algoritma – algoritma yang lain telah diterapkan untuk memecahkan masalah – masalah pencarian jalur (*path-planning*). Xiao mengenalkan *adaptive evolutionary planner / navigator* menggunakan berbagai macam operator untuk mengembangkan dan memperbaiki jalur – jalur yang akan dilalui [5]. Metode perencanaan trayektori secara hierarki pada ruang *pseudo-potential* untuk proses perakitan telah dikembangkan. Metode ini secara dasar mempunyai dua permukaan yaitu *trajectory generator* dan *configuration generator* menggunakan algoritma *virus-evolutionary* genetika [6]-[8], [5]. Sekarang, berbagai macam sistem *adaptive robot* telah dikenalkan untuk beradaptasi dengan lingkungan yang dinamis.

Pembelajaran dengan bantuan (*reinforcement learning*) telah diterapkan untuk membangun sebuah agen yang memaksimalkan penggunaan yang diharapkan, dengan

memberikan hadiah (*reward*) atau hukuman (*punishment*) dari lingkungan. Yaitu, *reinforcement learning* tidak menggunakan sinyal pembelajaran secara langsung, tetapi menggunakan evaluasi umpan - balik yang diperoleh melalui interaksi dengan lingkungan. Pencarian akan menjadi sulit jika ruang keadaan berukuran besar. Jika menggunakan *lookup table*, ruang kerja tiga dimensi diuraikan ke dalam sel – sel  $N \times N \times N$  dimana  $N$  menunjukkan ukuran dari sel – sel pada masing – masing sumbu. Selanjutnya, pencarian yang berturut – turut dari kandidat trayektori sulit dilakukan karena dinamika robot lengan. Sehingga, evaluasi manusia diperlukan untuk mewujudkan robot lengan yang *human-friendly* yaitu pergerakan lengan dianggap aman dan efisien oleh manusia. Efisien dengan maksud, (1) trayektori yang dihasilkan mempunyai jarak terdekat, (2) trayektori bergerak sepanjang titik – titik yang diberikan, (3) trayektori jauh dari halangan, (4) trayektori mencapai obyek dengan fungsi energi minimum dan (5) trayektori bergerak dengan kecepatan motor tinggi atau rendah. Sehingga robot akan mudah digunakan dan mudah berinteraksi.

Robot akan memilih sebuah tingkah laku dan membuat suatu keputusan untuk beraksi dengan manusia. Karena itu, akan dikenalkan metode perencanaan trayektori menggunakan fungsi *state-value* dan interaktif algoritma genetika. Fungsi *state-value* digunakan untuk memperkirakan evaluasi manusia. Dalam proyek ini, jumlah kuadrat jarak antara dua konfigurasi, jumlah kuadrat beda diantara sudut masing – masing joint antara dua konfigurasi, dan jumlah dari evaluasi manusia menggunakan fungsi *state-value* telah dikenalkan. Robot *human-friendly* harus menghasilkan trayektori berbasis evaluasi manusia.

## II. PARTICLE SWARM OPTIMIZATION (PSO)

Adalah salah satu teknik pencarian dalam ilmu komputer yang didasarkan pada suatu populasi. Seperti teknik pencarian yang lain, proses inisialisasi PSO dimulai dengan populasi acak yang dinamakan partikel. Tetapi tidak seperti metode lain, setiap partikel PSO mempunyai komponen kecepatan. Partikel – partikel mempunyai kecepatan yang secara dinamis akan mengalami perubahan tergantung dari pengalaman partikel – partikel. Oleh karena itu, partikel – partikel akan mempunyai solusi yang semakin baik dalam daerah proses pencariannya. Persamaan PSO mempunyai tiga komponen, seperti pada persamaan (1). Komponen pertama adalah *momentum part*,

kecepatan *inertia* suatu partikel. komponen kedua adalah *cognitive part*, merupakan bagian pengalaman partikel, belajar dari pengalaman partikel itu sendiri. Dan komponen ketiga adalah *social part*, merupakan kerja sama antar partikel, belajar dari pengalaman partikel – partikel lain. Berikut ini adalah persamaan PSO.

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[pbest_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[gbest_j(t) - x_{ij}(t)] \quad \dots(1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad \dots\dots\dots(2)$$

$$-v_{max} \leq v_i \leq v_{max}$$

Dimana  $v_{ij}(t)$  adalah kecepatan partikel ke-i dalam dimensi j pada waktu ke-t,  $x_{ij}(t)$  adalah posisi partikel ke-i dalam dimensi j pada waktu ke-t,  $c_1, c_2$  adalah konstanta positif,  $r_{1j}, r_{2j}$  adalah dua fungsi acak yang mempunyai nilai 0 sampai dengan 1,  $pbest_{ij}(t)$  adalah posisi terbaik dalam suatu partikel. Sedangkan  $gbest_i(t)$  adalah posisi terbaik dalam suatu populasi.

Seleksi posisi terbaik atau mencari *pbest* dimulai dengan menghitung nilai *fitness* setiap posisi yang dilalui satu partikel. Dengan membandingkan nilai *fitness* setiap posisi yang dilalui partikel tertentu maka akan didapat *personal best position* suatu partikel.

```

For each particle i=1,...,n do
    // Set the personal best position
    If  $f(x_i(t)) > f(pbest_i(t))$  then
         $pbest_i(t) = x_i(t)$ 
    End

```

**End**

Dimana  $f()$  adalah fungsi nilai *fitness*. Seleksi posisi terbaik dalam populasi dimulai dengan membandingkan nilai – nilai *fitness*  $pbest_i(t)$  dengan nilai *fitness*  $gbest_i(t)$ , jika nilai *fitness*  $pbest_i(t)$  kurang dari nilai *fitness*  $gbest_i(t)$  maka ganti  $gbest_i(t)$  dengan  $pbest_i(t)$ . berikut adalah algoritma seleksi tersebut.

```

For each particle i=1,...,n do
    // Set the global best position
    If  $f(pbest_i(t)) > f(gbest_i(t))$  then
         $pbest_i(t) = gbest_i(t)$ 
    End

```

**End**

Update kecepatan dan posisi digambarkan sebagai berikut.

```

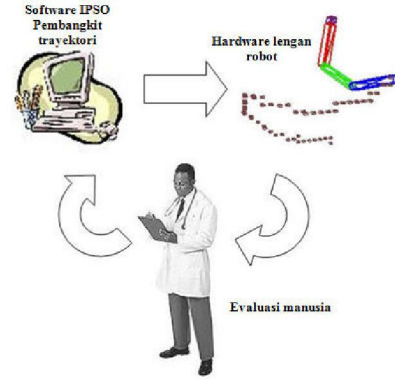
For each particle i=1,...,n do
    Update the velocity
    Update the position
End

```

jika kecepatan partikel melebihi  $\pm v_{max}$  maka kecepatan akan menjadi  $\pm v_{max}$ .

### III. RANCANGAN SISTEM

Trayektori yang dibangkitkan oleh CPU akan dikirim ke rangkaian *hardware* dan melalui evaluasi manusia akan dinilai trayektori – trayektori yang terbaik pada saat itu. Trayektori yang terbaik akan tergantung dari operator. Gambar berikut adalah blok diagram keseluruhan sistem.



Gambar 1. Blok diagram sistem.

Nilai *fitness* merupakan ukuran yang menentukan seberapa baik setiap individu yang dibangkitkan. Tujuan utama dari keseluruhan proses ini adalah untuk membangkitkan trayektori yang mempunyai jarak minimum dari konfigurasi awal sampai konfigurasi akhir dan mempunyai nilai evaluasi manusia tertinggi.

$$fit_i = w_1 f_p + w_2 f_d + w_3 f_v \quad \dots\dots\dots(3)$$

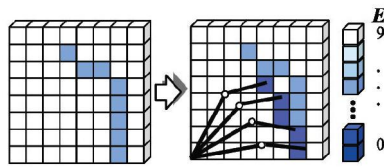
$$f_v = \sum_{(x,y,z) \in P} value(x, y, z) \quad \dots\dots\dots(4)$$

$w_1, w_2, w_3$  adalah koefisien jarak,  $f_p$  adalah jumlah kuadrat dari jarak antara 2 konfigurasi,  $f_d$  adalah jumlah kuadrat dari sudut antara 2 konfigurasi pada masing – masing *joint*, dan  $f_v$  adalah jumlah nilai evaluasi manusia yang menggunakan fungsi *state-value*.

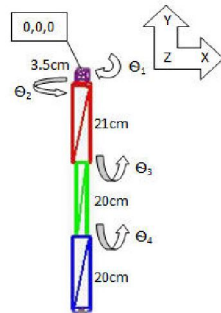
Evaluasi manusia yang diberikan akan digunakan untuk memperbarui fungsi *state-value*. Gambar 2 menunjukkan contoh cara pemberian nilai dan cara memperbarui setiap trayektori yang dihasilkan, dengan memisalkan kotak berbentuk dua dimensi.

$$value(x, y, z) \leftarrow (1 - \gamma) value(x, y, z) + \gamma E \quad \dots\dots\dots(5)$$

$\gamma$  adalah ukuran step,  $E$  adalah evaluasi manusia,  $value(x, y, z)$  adalah nilai pada koordinat x, y, z.

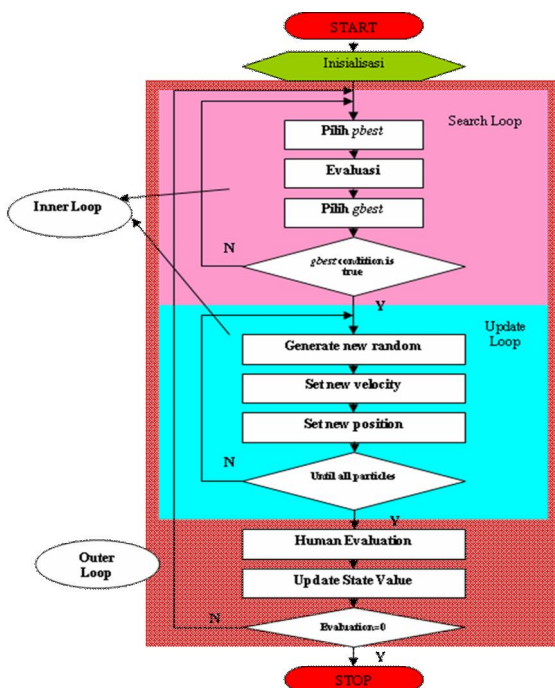


Gambar 2. Sel – sel fungsi state-value.



Gambar 3. Pergerakan dan ukuran robot lengan.

Gambar 3 adalah pergerakan dan ukuran lengan robot yang nanti digunakan dalam percobaan. Titik awal robot adalah titik (0,0,0), memanjang ke arah sumbu y negatif. Gambar berikut adalah *flowchart* dari *interactive particle swarm optimization* (IPSO).



Gambar 4. Flowchart IPSO

Terdapat dua bagian utama dalam *flowchart* diatas yaitu *inner-loop* dan *outer-loop*. Pada *inner-loop* terdapat proses PSO sedangkan *human-evaluation* dan *update state-value* terdapat di *outer-loop*. Proses pertama adalah inisialisai

dimana kandidat – kandidat atau partikel – partikel solusi akan dibangkitkan secara acak. Dari setiap kandidat tersebut akan dicari nilai pbest (pengalaman terbaik menurut suatu partikel) kemudian mengevaluasi trayektori partikel tersebut melalui fungsi evaluasi *fitness*. Kemudian mencari gbest (pengalaman terbaik dalam suatu populasi), proses tersebut berlanjut sampai seluruh partikel diproses.

Proses selanjutnya adalah memperbarui posisi masing – masing partikel. Pertama adalah menghasilkan bilangan acak yang digunakan dalam menghitung persamaan PSO. Kemudian memperbarui kecepatan dan posisi masing – masing partikel. Proses tersebut berlangsung sampai seluruh partikel telah diproses. Evaluasi manusia berlangsung dalam tahap ini, *operator* memberikan nilai pada trayektori terbaik yang telah dibangkitkan.

#### IV. HASIL PERCOBAAN

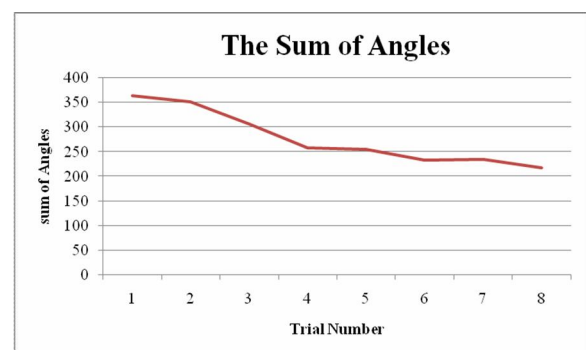
Pengujian lengan robot dilakukan pada beberapa parameter yaitu.

TABEL I  
PARAMETER IPSO YANG DIBUAT KONSTAN

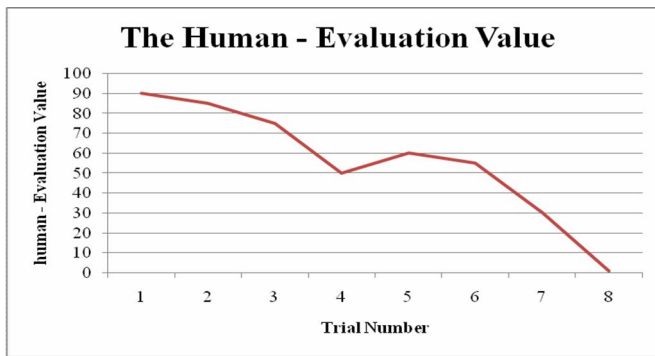
Population Size	50
Num Configuration	10
Local Evaluations (Inner loop times)	25
Space Size	22
PSO Cognitive Constant	0.005
PSO Social Constant	0.01

TABEL II  
PARAMETER IPSO YANG DIBUAT VARIABEL

Coba ke	SUM OF ANGLE	SUM OF DISTANCES	SUM OF ESTIMATED EVALUATION
1.	0,8	0,1	0,1
2.	0,1	0,8	0,1
3.	0,1	0,1	0,8

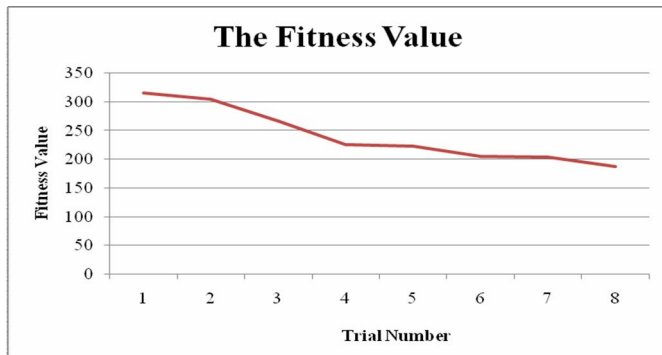


Gambar 5. Grafik *sum-of-angle*,  $w_1=0.8$ ,  $w_2=0.1$ ,  $w_3=0.1$ .



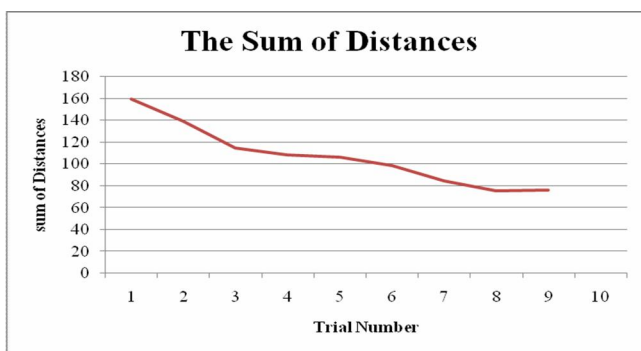
Gambar 6. Grafik nilai evaluasi manusia,  $w1=0.8$ ,  $w2=0.1$ ,  $w3=0.1$ .

Dari gambar 7 nampak nilai fitness mengalami penurunan (trayektori semakin baik) dan grafik jumlah sudut yang semakin baik.

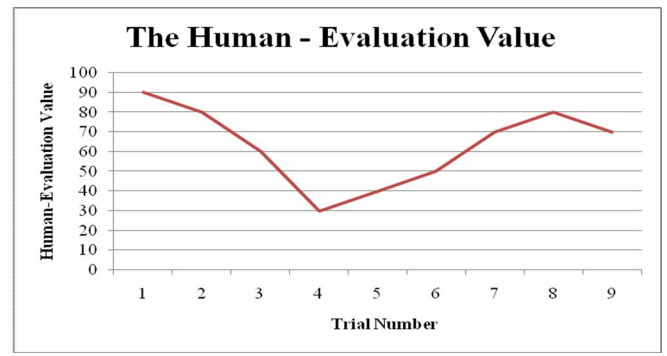


Gambar 7. Grafik nilai *fitness* 1,  $w1=0.8$ ,  $w2=0.1$ ,  $w3=0.1$ .

Tetapi saat trial kelima gambar 6 grafik evaluasi manusia mengalami kenaikan nilai yang berarti trayektori tersebut kurang disenangi oleh manusia. Gerakan atau trayektori yang dihasilkan hanya tergantung dari jumlah sudut terkecil. Namun terlihat trayektori mencapai nilai evaluasi manusia terbaik saat trial ke-8.

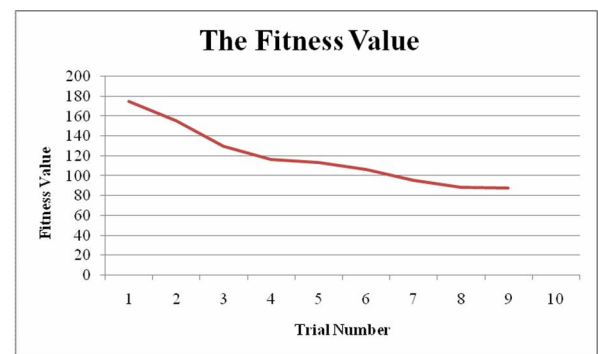


Gambar 8. Grafik *sum of distance*,  $w1=0.1$ ,  $w2=0.8$ ,  $w3=0.1$ .



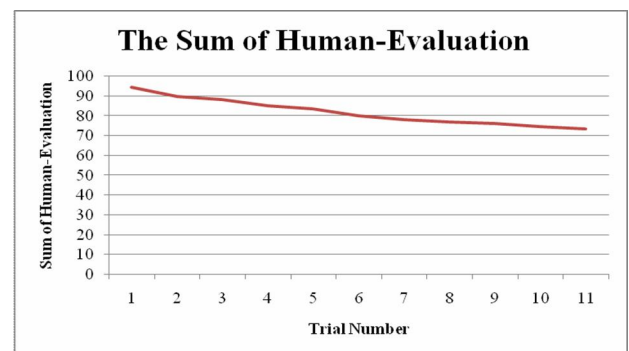
Gambar 9. Grafik nilai evaluasi manusia,  $w1=0.1$ ,  $w2=0.8$ ,  $w3=0.1$ .

Gambar 8 dan 9 menunjukkan perbaikan nilai tetapi saat trial kelima grafik evaluasi manusia mengalami kenaikan, gerakan robot menjadi tak menentu atau acak. Gerakan yang dibangkitkan hanya mengkondisikan jumlah jarak yang terkecil. Nilai evaluasi manusia tidak pernah mencapai solusi terbaik.



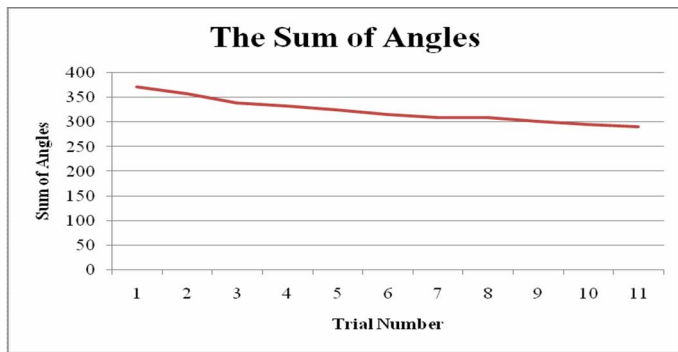
Gambar 10. Grafik nilai *fitness*,  $w1=0.1$ ,  $w2=0.8$ ,  $w3=0.1$ .

Gambar 10 grafik nilai *fitness*, meskipun menunjukkan perbaikan nilai tetapi masih evaluasi menunjukkan trayektori yang buruk saat *trial* kelima.



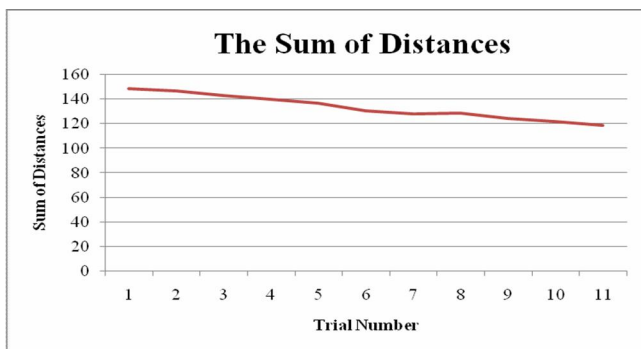
Gambar 11. Grafik *sum of human-evaluation*,  $w1=0.1$ ,  $w2=0.1$ ,  $w3=0.8$ .





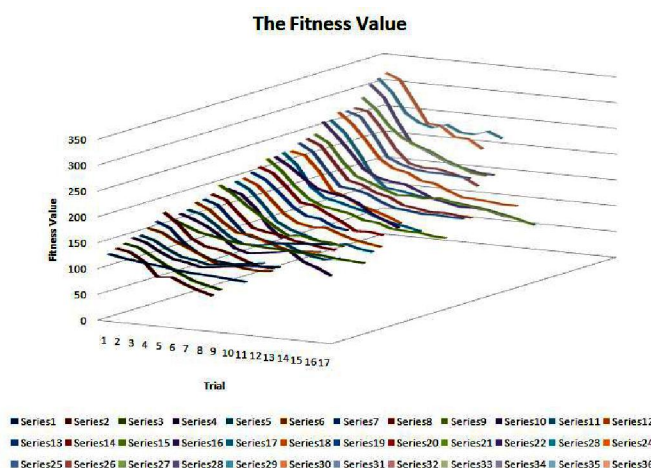
Gambar 12. Grafik *sum of angles*,  $w_1=0.1$ ,  $w_2=0.1$ ,  $w_3=0.8$ .

Gambar 11 dan 12 keseluruhan grafik mengalami perbaikan tetapi dalam skala yang kecil, terlihat sampai trial ke-11 tetap tidak bisa mencapai solusi terbaik. Trayektori yang dibangkitkan seperti terperangkap dalam daerah tertentu.

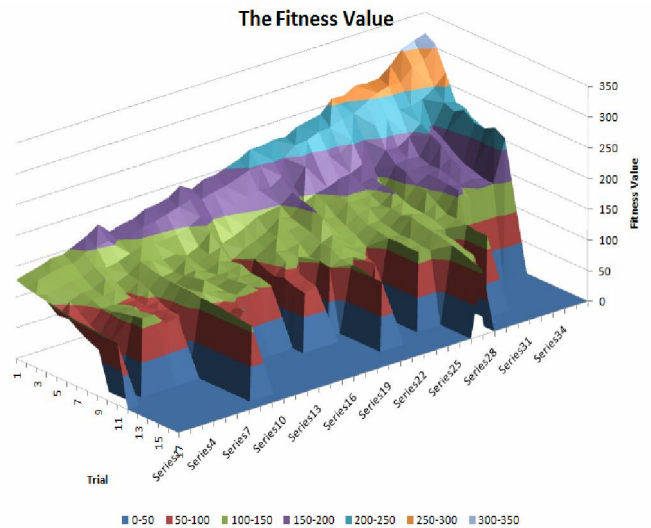


Gambar 13. Grafik *sum of distances*,  $w_1=0.1$ ,  $w_2=0.1$ ,  $w_3=0.8$ .

Gambar 11, 12, dan 13, meskipun menunjukkan keseluruhan grafik yang semakin baik (grafik menurun) tetapi kurang bisa untuk menemukan solusi yang diharapkan ditunjukkan dengan grafik nilai evaluasi manusia yang datar. Hal ini menunjukkan gerakan robot seperti terperangkap dalam daerah tertentu.



Gambar 14. Grafik nilai *fitness*, pop size=50.



Gambar 15. Grafik 3D nilai *fitness*, pop size=50.

TABEL III  
PARAMETER PEMBERAT PERCOBAAN IPSO

No	weight			Trial	No	weight			Trial
	Series	w1	w2	w3		Series	w1	w2	w3
Series 1	0.1	0.1	0.8	11↑	Series 20	0.3	0.5	0.2	10
Series 2	0.1	0.2	0.7	8	Series 21	0.3	0.6	0.1	14
Series 3	0.1	0.3	0.6	8					
Series 4	0.1	0.4	0.5	10↑	Series 22	0.4	0.1	0.5	10
Series 5	0.1	0.5	0.4	10↑	Series 23	0.4	0.2	0.4	11
Series 6	0.1	0.6	0.3	10↑	Series 24	0.4	0.3	0.3	9
Series 7	0.1	0.7	0.2	10↑	Series 25	0.4	0.4	0.2	13↑
Series 8	0.1	0.8	0.1	9↑	Series 26	0.4	0.5	0.1	13↑
Series 9	0.2	0.1	0.7	15↑	Series 27	0.5	0.1	0.4	17
Series 10	0.2	0.2	0.6	12	Series 28	0.5	0.2	0.3	15
Series 11	0.2	0.3	0.5	12↑	Series 29	0.5	0.3	0.2	11
Series 12	0.2	0.4	0.4	10↑	Series 30	0.5	0.4	0.1	14
Series 13	0.2	0.5	0.3	10↑					
Series 14	0.2	0.6	0.2	10	Series 31	0.6	0.1	0.3	10↑
Series 15	0.2	0.7	0.1	10↑	Series 32	0.6	0.2	0.2	10↑
					Series 33	0.6	0.3	0.1	10↑
Series 16	0.3	0.1	0.6	5					
Series 17	0.3	0.2	0.5	11	Series 34	0.7	0.1	0.2	10↑
Series 18	0.3	0.3	0.4	11	Series 35	0.7	0.2	0.1	10↑
Series 19	0.3	0.4	0.3	8					
					Series 36	0.8	0.1	0.1	8

Data *series* 16 sampai 21 dapat mencapai trayektori yang terbaik terutama *series* 16, yaitu mencapai trayektori terbaik saat *trial* ke-5. Data tersebut dianggap terbaik menurut subjektif manusia sebagai lawan jabat tangan. Data *series* 16 mempunyai nilai konstanta  $w_1 = 0.3$ ,  $w_2 = 0.1$ ,  $w_3 = 0.6$ , hal ini berarti trayektori yang dibangkitkan lebih mengutamakan evaluasi manusia dengan sebagian jumlah sudut dan sebagian kecil lagi jumlah jarak.

Data *series* 16 sampai 21 menunjukkan data terbaik dengan jumlah *trial* yang sedikit jika dibandingkan dengan data – data yang lain. Hal ini berarti data optimum diperoleh saat konstanta  $w_1 = 0.3$ .

*Series* 22 sampai 30 dapat mencapai trayektori yang terbaik meskipun dengan jumlah *trial* yang cukup tinggi, namun saat nilai konstanta ditingkatkan melalui *series* 31 sampai 36 maka trayektori tidak dapat diperoleh. Trayektori yang dibangkitkan saat *series* 31 sampai 36 (nilai  $w_1$  cukup tinggi dibanding parameter lain) mempunyai gerakan yang halus, namun semakin sulit untuk menemukan solusi trayektori – trayektori yang dianggap terbaik oleh user

## V. KESIMPULAN

Hasil – hasil percobaan menunjukkan beberapa hal yaitu (1) Nilai pemberat  $w_1$  (pemberat *sum of angle*) diatas 0.6 mengakibatkan gerakan robot yang kurang baik menurut evaluasi manusia, gerakan lengan robot menjadi lebih halus namun tidak menunjukkan jabat tangan yang baik, (2) Nilai pemberat  $w_2$  (pemberat *sum of distance*) diatas 0.4 dengan  $w_1$  kurang dari 0.3 mengakibatkan gerakan robot menjadi acak atau tak menentu atau bahkan melebihi batas – batas jabat tangan yang diperbolehkan, (3) Nilai pemberat  $w_3$  (pemberat *sum of human-evaluation*) diatas 0.6 mengakibatkan robot terperangkap pada daerah tertentu saja, tidak bisa menemukan daerah solusi yang terbaik, (4) Perpaduan ketiga parameter  $w_1, w_2, w_3$  akan menemukan solusi yang terbaik (yaitu saat  $w_1 = 0.3, w_2 = 0.1, w_3 = 0.6$ ), (5) Terlihat pada data hasil percobaan bahwa IPSO dapat menemukan trayektori lebih cepat (yaitu dengan jumlah *trial* 5) daripada IGA (yaitu dengan jumlah *trial* 10).

## DAFTAR PUSTAKA

- [1] O.Khatib, K.Yokoi, O.Brock, K-S.Chang, and A.Casal, "Robots in Human Environments", In Archives of Control Sciences, *Special Issue on Granular Computing*. Vol.11 (XLVII), No. 3-4, pp.123-128, 2002.
- [2] N.Kubota, D.Hisajima, F.Kojima, and T.Fukuda, "Fuzzy and Neural Computing for Communication of a Partner Robot", *Journal of Multi-Valued Logic and Soft Computing*, Vol.9, No.2, pp.221-239, 2003.
- [3] S.Kristensen, S.Horstmann, J.Klandt, F.Lohnert and A.Stopp, "Human-Friendly Interaction for Learning and Cooperation", *Proc. of the 2001 IEEE ICRA*, pp.2590-2595, 2001.
- [4] Artur M. Arsenio, Embodied Vision - Perceiving Objects from Actions, *Proceeding of the 12th IEEE Workshop Robot and Human Interactive Communication RO-MAN 2003*, pp.365-371, 2003.
- [5] J.Xiao, Z.Michalewicz, L.Zhang, K.Trojanowski, "Adaptive Evolutionary Planner/Navigator for Mobile Robots", *IEEE Trans. Evolutionary Computation*, Vol.1, No.1, 1998, pp.18-28.
- [6] T.Fukuda, N.Kubota, and T.Arakawa, "GA Algorithms in Intelligent Robot", *Fuzzy Evolutionary Computation*, Kluwer Academic Publishers, 1997, pp.81-105.
- [7] T.Arakawa and T.Fukuda, "Natural Motion Trajectory Generation of Biped Locomotion Robot Using Genetic Algorithm through Energy Optimization", in *Proc. of The IEEE International Conference on Systems, Man, and Cybernetics*, 1996, pp.1495-1500.

- [8] N.Baba and N.Kubota, "Collision Avoidance Planning of a Robot Manipulator by Using Genetic Algorithm-A Consideration for the Problem in which Moving Obstacles and/or Several Robots Are in the Workspace", *Proc. of The First IEEE Conference on Evolutionary Computation*, 1994, pp.714-719.
- [9] J.H. Holland, "Adaptation in Natural and Artificial Systems", Cambridge, Massachusetts, MIT Press, 1995.
- [10] S.J.Russell, P.Norvig, "Artificial Intelligence", Prentice Hall, Inc., 1995.
- [11] N.Kubota, I.Adji S., F.Kojima, "Interactive Genetic Algorithm for Trajectory Generation of a Robot Manipulator", *Proceeding (CD ROM) of 4 th Asia-Pasific Conference On Simulated Evolution And Learning (SEALO2)*, pp.146-150, Singapore, 18-22 November, 2002.
- [12] Shi Yuhui, "Particle Swarm Optimization", *IEEE Neural Networks Society*, February 2004.
- [13] A.P. Engelbrecht, "Fundamentals of Computational Swarm Intelligence", John Wiley & Sons, 2005.